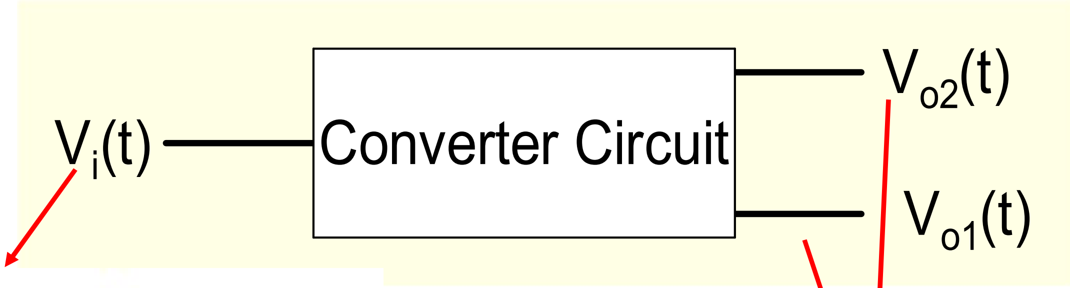
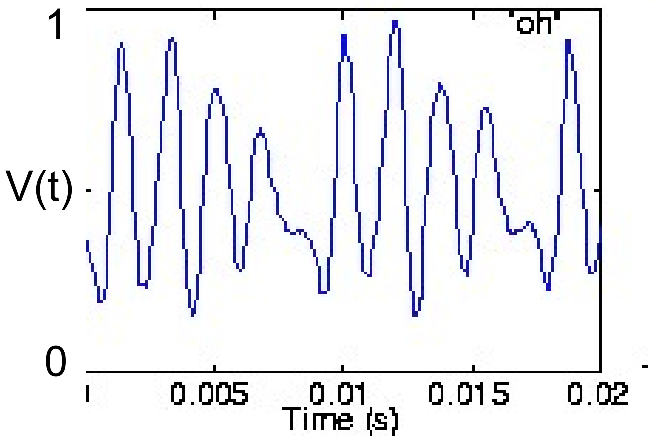
# ESC201T : Introduction to Electronics

## Lecture 31: Digital Circuits-1

B. Mazhari
Dept. of EE, IIT Kanpur

# Analog vs. Digital Signal




Can take only discrete values (0,1,2..9)

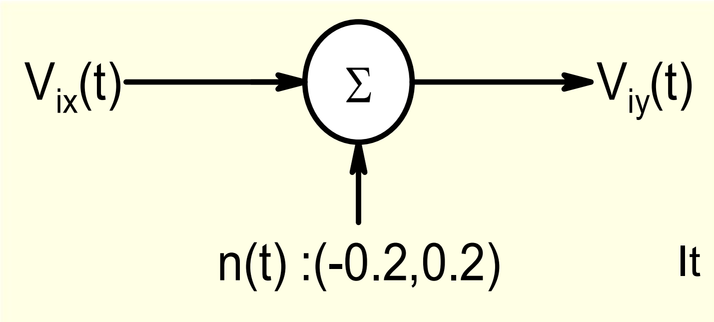
| Analog Voltage (V) | Vo2 (V) | Vo1 (V) |
|---|---|---|
| 0.45 | 4 | 5 |
| 0.75 | 7 | 5 |
| 0.93 | 9 | 3 |
| 0.932 | 9 | 3 |
| 0.934 | 9 | 3 |
| 0.936 | 9 | 4 |

Can take any value in the range 0-1V

Loss in accuracy

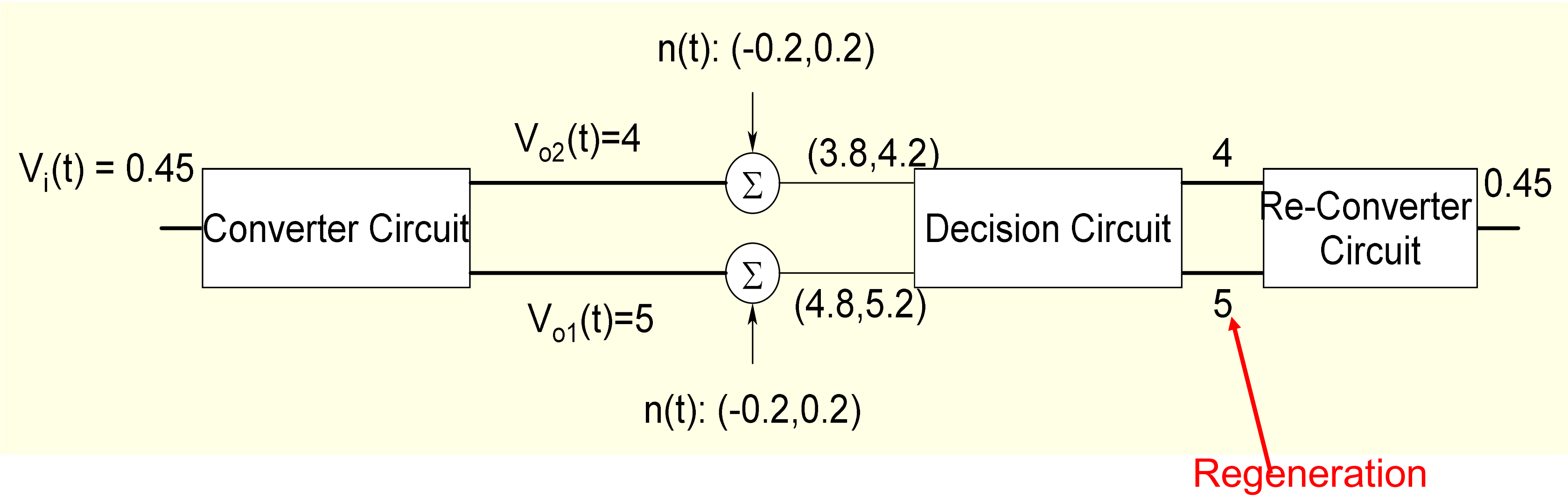
## Advantages of using digital Signals

**Robustness towards noise**



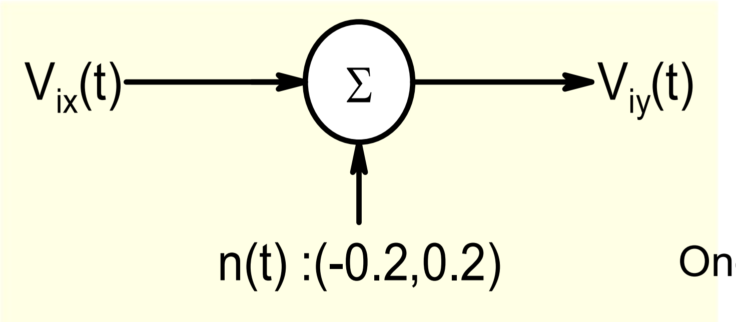

$$V_{ix}(t) = 0.45 \rightarrow V_{iy}(t):(0.25, 0.65)$$

It is very difficult to recover the original signal



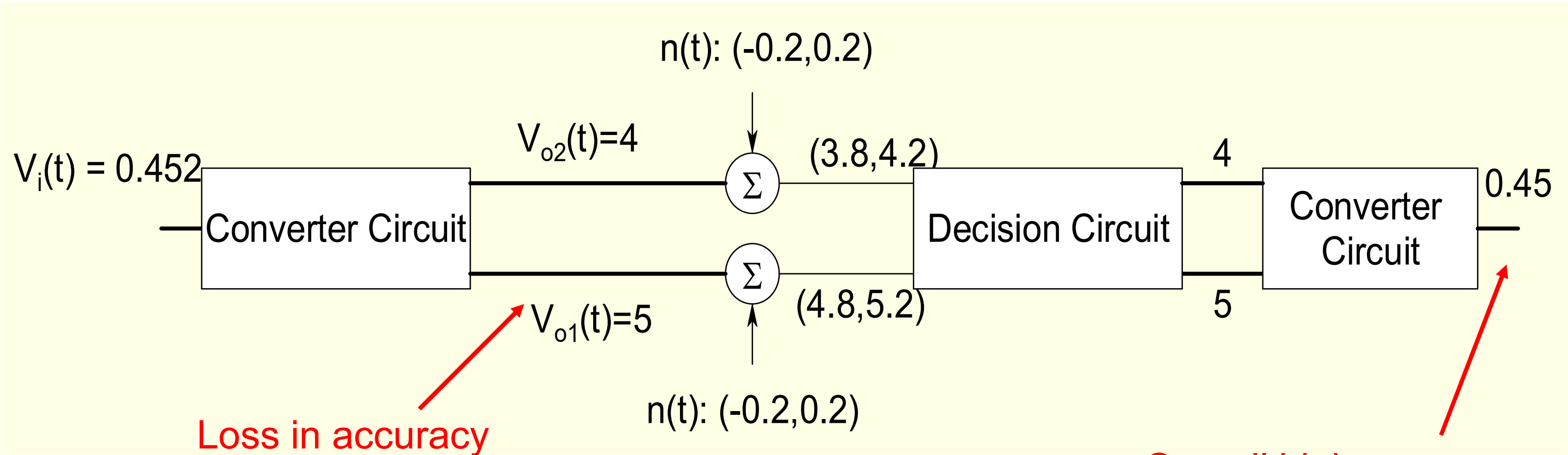
**Advantages of using digital Signals**

**Robustness towards noise**

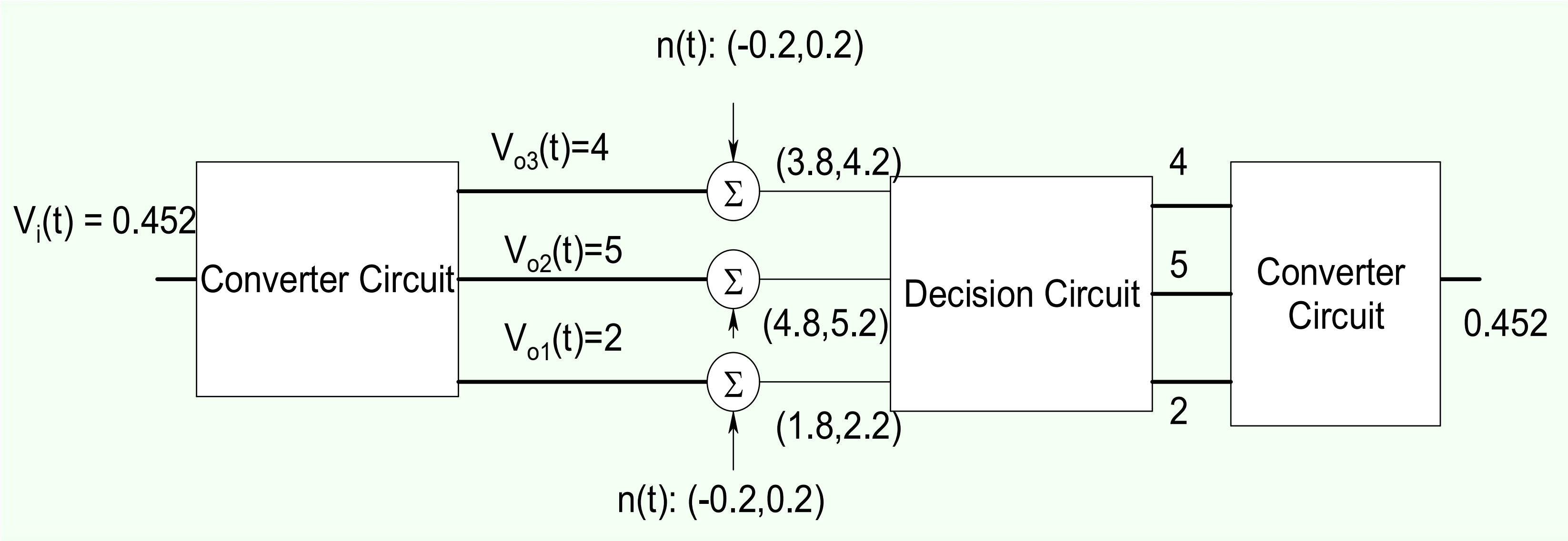



$$V_{ix}(t) = 0.452 \rightarrow V_{iy}(t) : (0.252, 0.652)$$

One is uncertain about the first decimal place !!



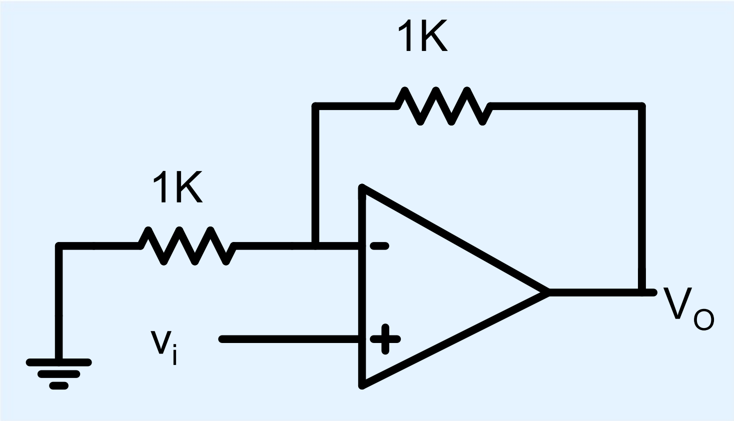

Overall higher accuracy




One can get the desired accuracy using larger number of digits

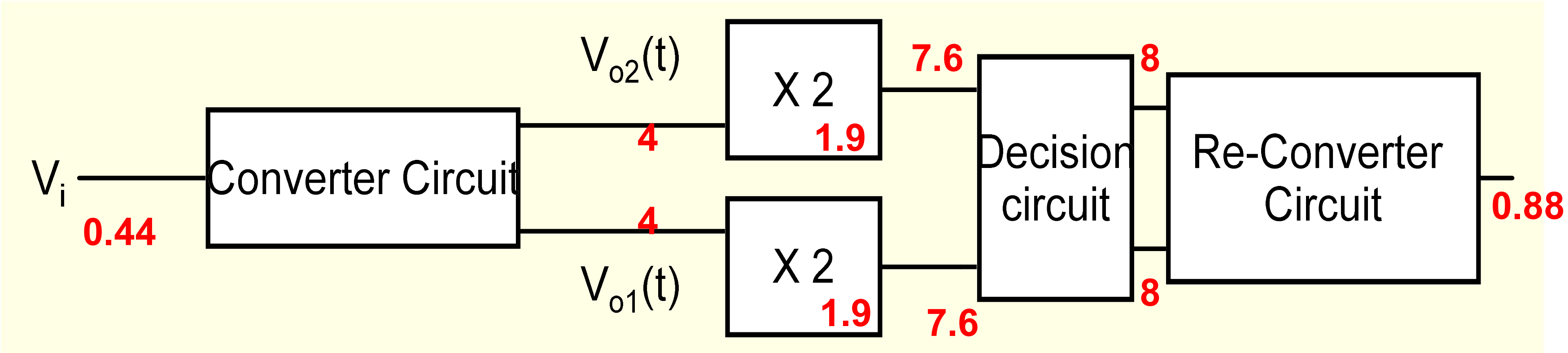
## Accurate Processing ?

Suppose we would like to multiply a signal by a factor of 2




Because of tolerances etc, we would not get a gain of 2. Suppose the gain is 1.9.

For Vi = 0.44, we would get 0.836 instead of 0.88V.

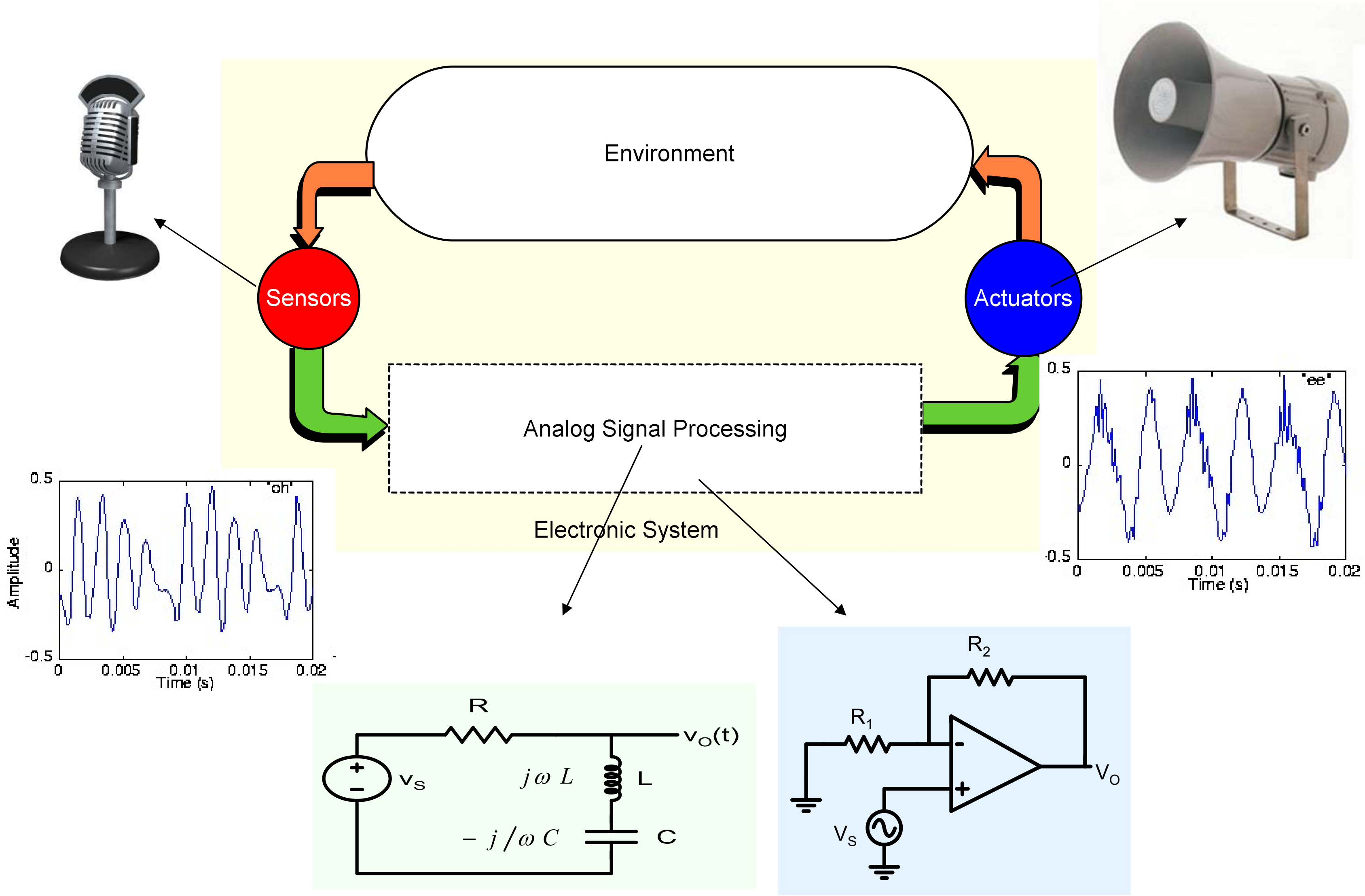



**Processing of digital signals is often much simpler if numbers are represented properly !**

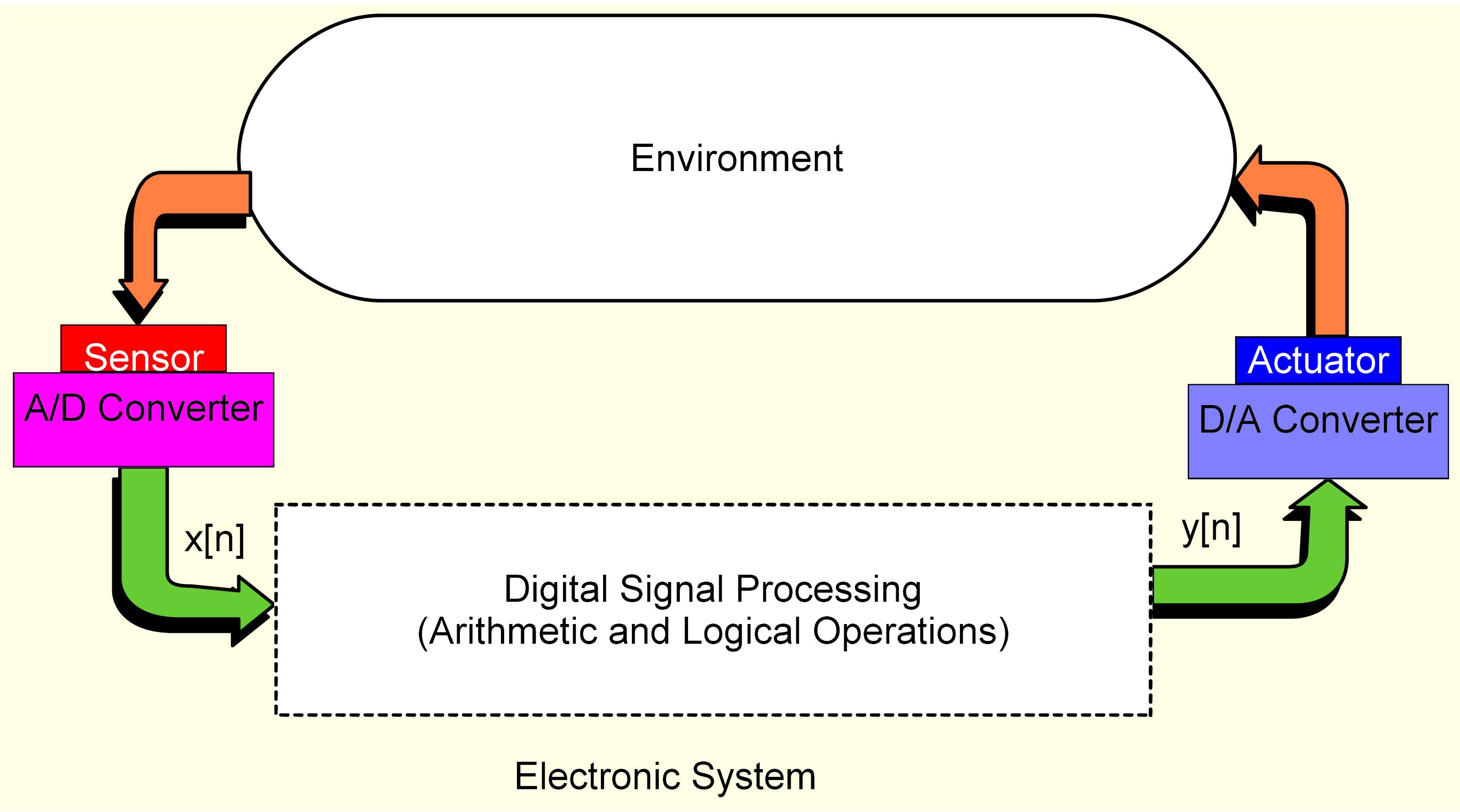
## Digital circuits allow much more complex information processing

**Deep Blue** was a chess-playing computer developed by IBM. On May 11, 1997, the machine won a six-game match by two wins to one with three draws against world champion Garry Kasparov Kasparov accused IBM of cheating and demanded a rematch, but IBM declined and dismantled Deep Blue. Kasparov had beaten a previous version of Deep Blue in 1996….wikipedia

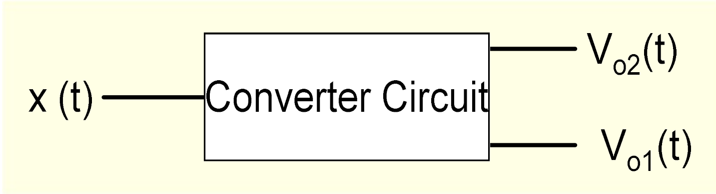
# Analog signal Processing



## Digital signal Processing

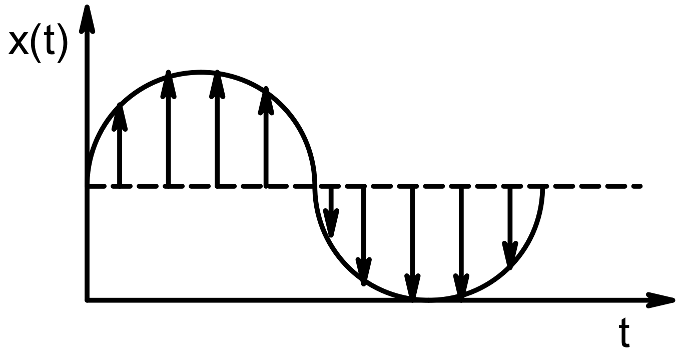


## Converting signals into a sequence of numbers and vice-versa



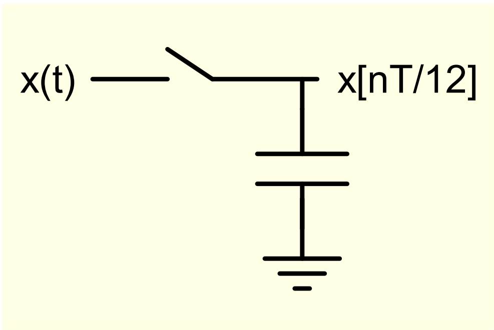

$$x(t) = 0.5 + 0.5 \times \sin(\frac{2\pi}{10^{-3}} t)$$

## Step-1: Sampling

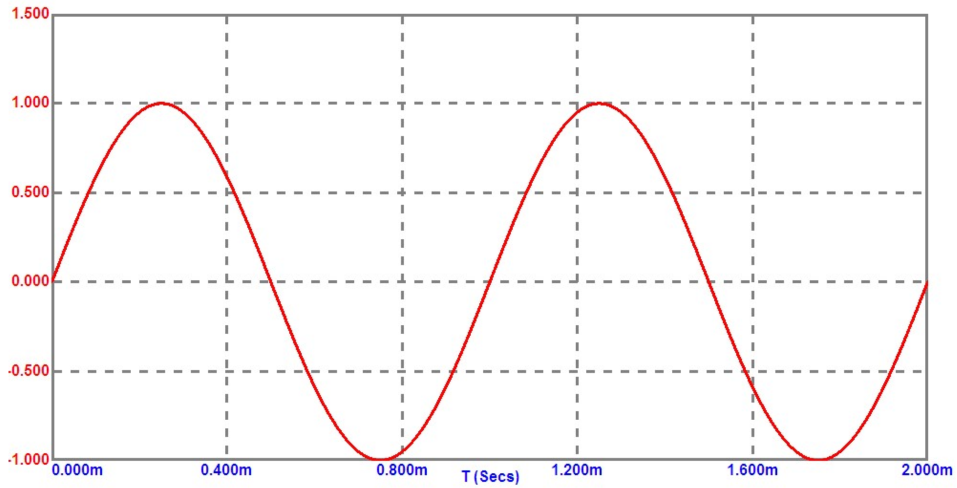


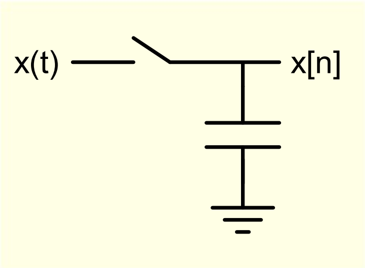

Sample at intervals of T/12

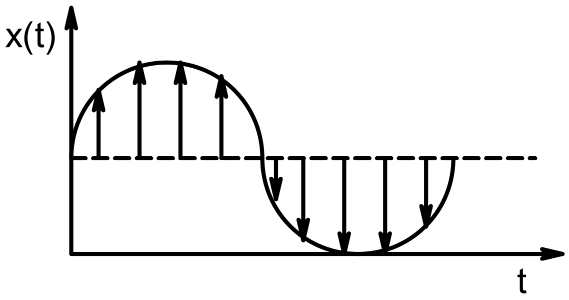
$$x[n\frac{T}{12}] = [0.5, 0.75, 0.93, 1, 0.93, 0.75, 0.5, 0.25, 0.067, 0, 0.067, 0.25, 0.5....]$$

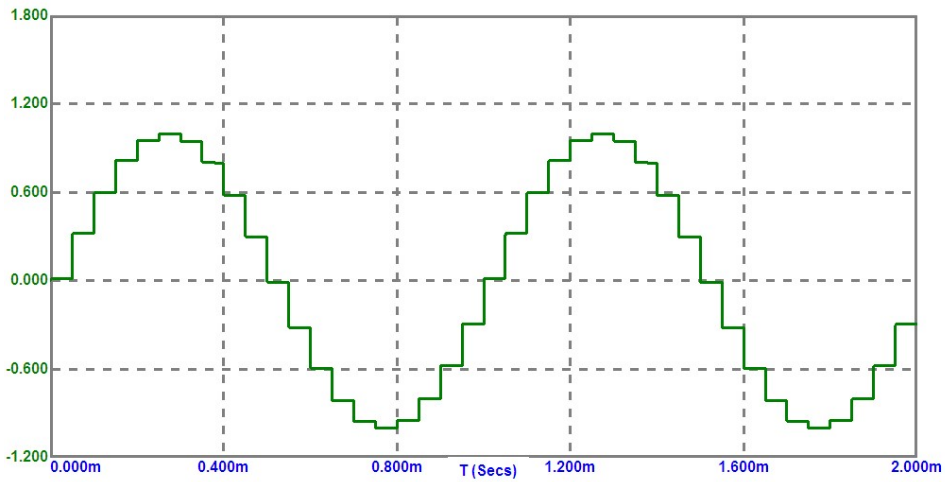


# Example



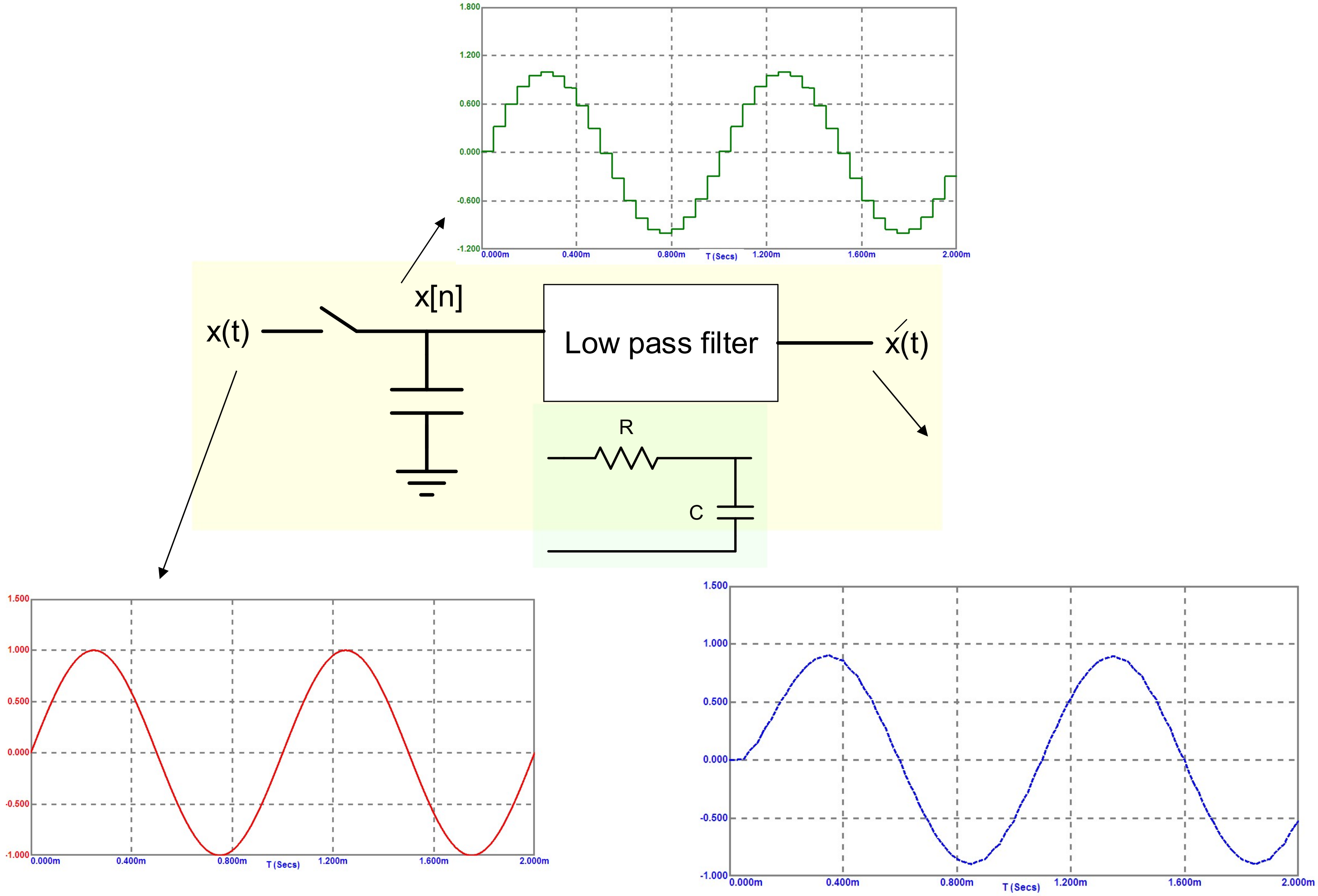



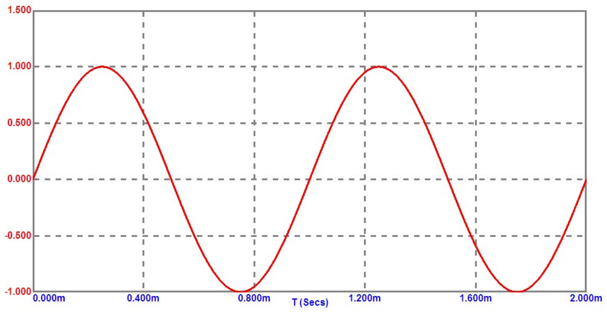



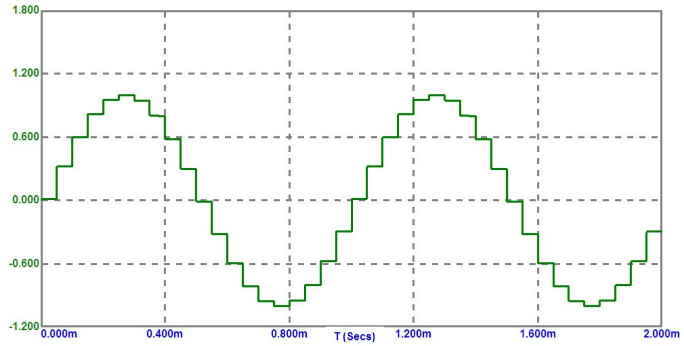

Sample and Hold

One can recover the original waveform by low pass filtering the sampled waveform

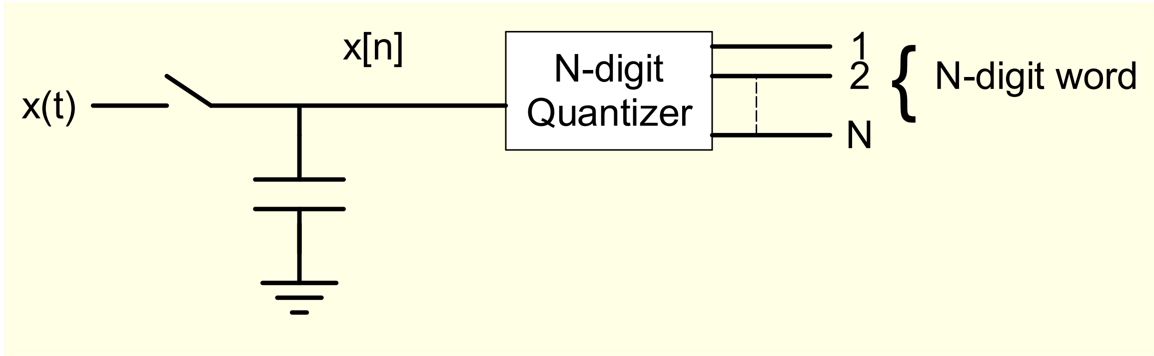


# Converting sampled waveform into a sequence of numbers



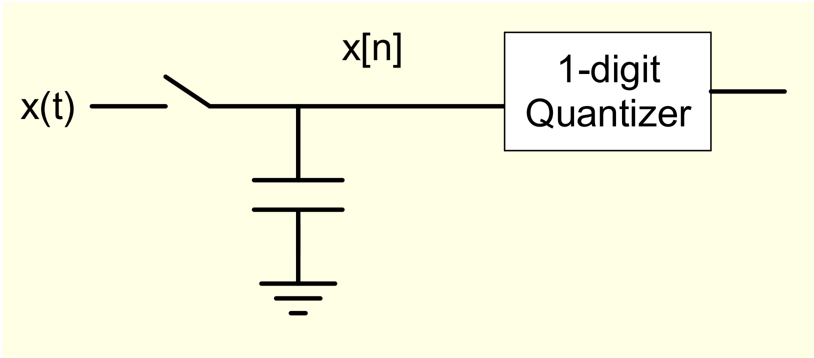

## Quantization:

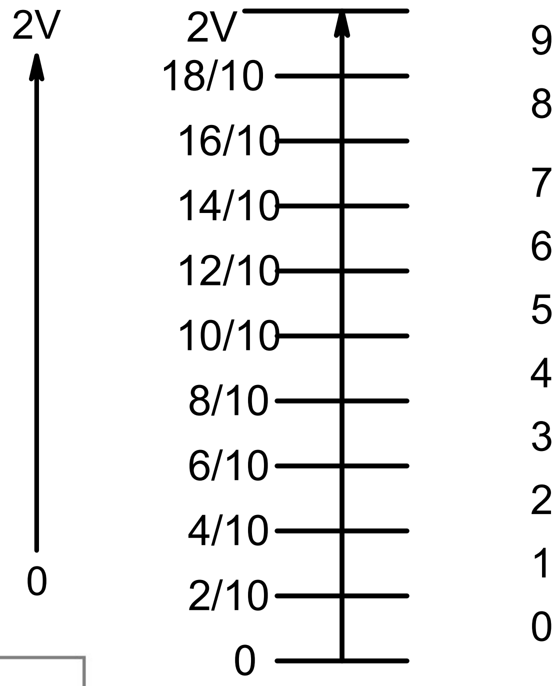


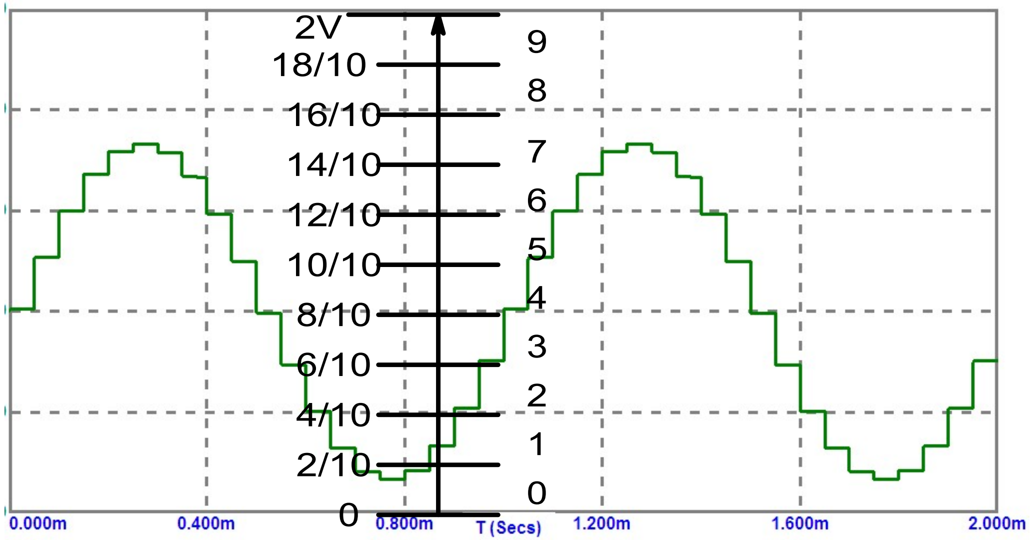

1-digit quantizer: [0,1,3,5,4,3,2,0,1,3,5…….]

2-digit quantizer: [00,12,32,57,42,31,22,00,12,32,57…….]

## Quantization:




It is obvious that quantization introduces errors !

## Quantization:

$$x(t) = 0.5 + 0.5 \times \sin(\frac{2\pi}{10^{-3}} t)$$

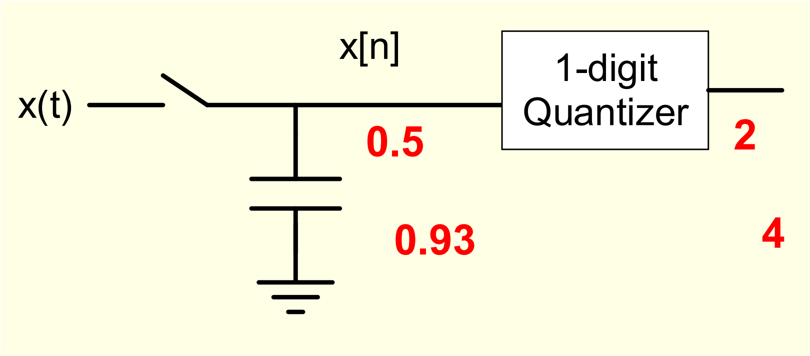
Sample at intervals of T/12

$$x[n\frac{T}{12}] = [0.5, 0.75, 0.93, 1, 0.93, 0.75, 0.5, 0.25, 0.067, 0, 0.067, 0.25, 0.5....]$$

| Voltage Range (Volts) | Number |
|---|---|
| 0-0.2 | 0 |
| 0.2-0.4 | 1 |
| 0.4-0.6 | 2 |
| 0.6-0.8 | 3 |
| 0.8-1.0 | 4 |
| 1.0-1.2 | 5 |
| 1.2-1.4 | 6 |
| 1.4-1.6 | 7 |
| 1.6-1.8 | 8 |
| 1.8-2.0 | 9 |

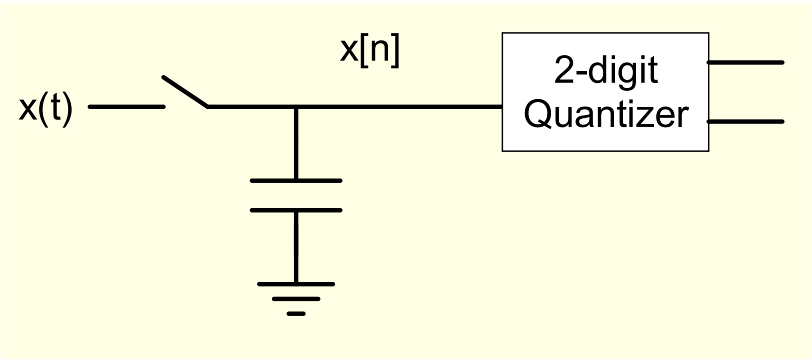
| Analog Voltage | Digital voltage or Number |
|---|---|
| 0.5 | 2 |
| 0.75 | 3 |
| 0.93 | 4 |
| 1 | 5 |
| 0.93 | 4 |
| 0.75 | 3 |
| 0.5 | 2 |
| 0.25 | 1 |
| 0.067 | 0 |
| 0 | 0 |
| 0.067 | 0 |
| 0.25 | 1 |
| 0.5 | 2 |

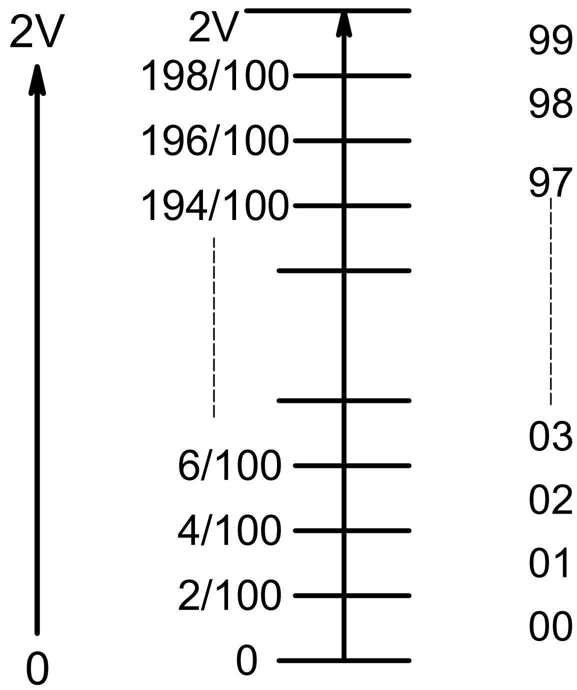
$$x(t) = 0.5 + 0.5 \times \sin(\frac{2\pi}{10^{-3}} t)$$




| Analog Voltage | Digital voltage or Number |
|---|---|
| 0.5 | 2 |
| 0.75 | 3 |
| 0.93 | 4 |
| 1 | 5 |
| 0.93 | 4 |
| 0.75 | 3 |
| 0.5 | 2 |
| 0.25 | 1 |
| 0.067 | 0 |
| 0 | 0 |
| 0.067 | 0 |
| 0.25 | 1 |
| 0.5 | 2 |

## 2-digit Quantization:



## 2-digit Quantization:

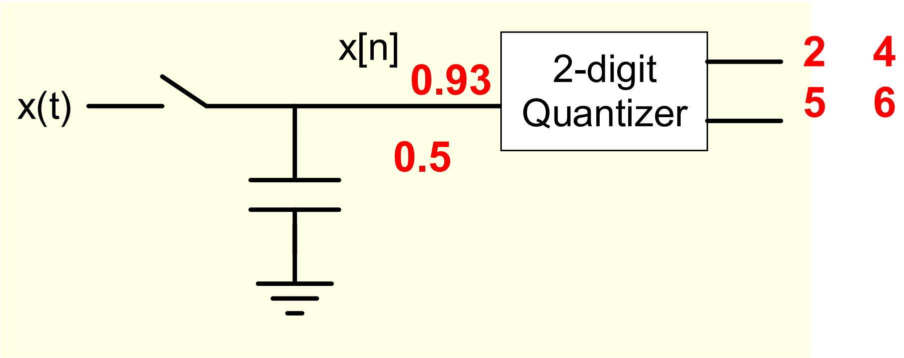
$$x(t) = 0.5 + 0.5 \times \sin(\frac{2\pi}{10^{-3}} t)$$

Sample at intervals of T/12

| Voltage Range (Volts) | Number |
|---|---|
| 0-0.02 | 00 |
| 0.02-0.04 | 01 |
| 0.04-0.06 | 02 |
| - | - |
| - | - |
| - | - |
| 1.92-1.94 | 96 |
| 1.94-1.96 | 97 |
| 1.96-1.98 | 98 |
| 1.98-2.0 | 99 |

| Analog Voltage | Number |
|---|---|
| 0.5 | 25 |
| 0.75 | 37 |
| 0.93 | 46 |
| 1 | 50 |
| 0.93 | 46 |
| 0.75 | 37 |
| 0.5 | 25 |
| 0.25 | 12 |
| 0.067 | 03 |
| 0 | 00 |
| 0.067 | 03 |
| 0.25 | 12 |
| 0.5 | 25 |

$$x(t) = 0.5 + 0.5 \times \sin(\frac{2\pi}{10^{-3}} t)$$

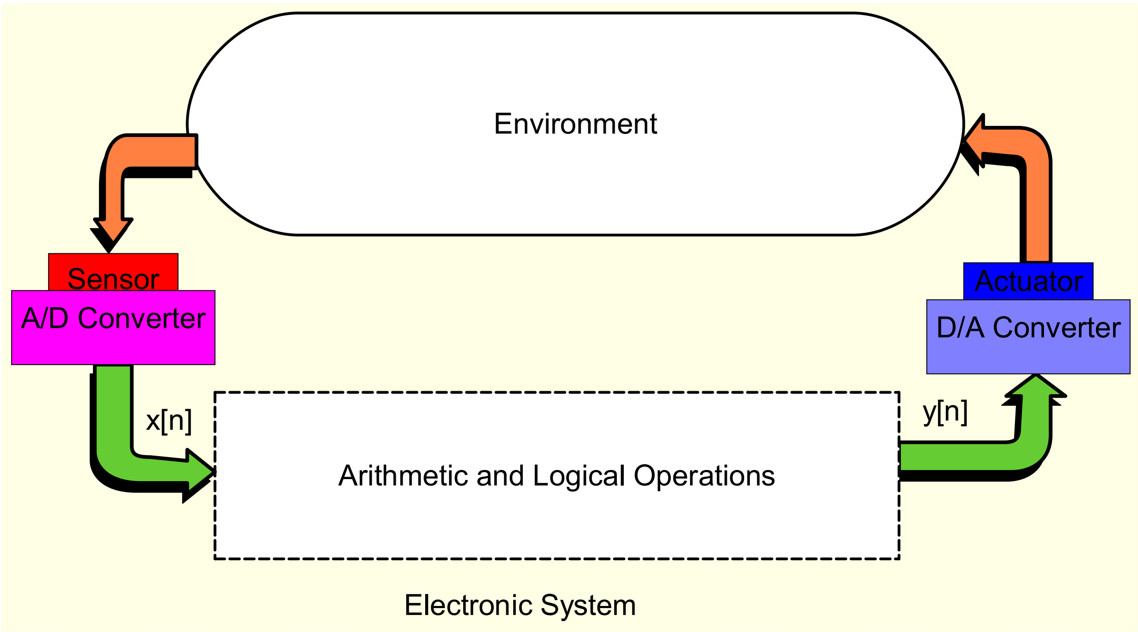



| Analog Voltage | Number |
|---|---|
| 0.5 | 25 |
| 0.75 | 37 |
| 0.93 | 46 |
| 1 | 50 |
| 0.93 | 46 |
| 0.75 | 37 |
| 0.5 | 25 |
| 0.25 | 12 |
| 0.067 | 03 |
| 0 | 00 |
| 0.067 | 03 |
| 0.25 | 12 |
| 0.5 | 25 |

## Converting numbers back to signals

$$x(t) = 0.5 + 0.5 \times \sin(\frac{2\pi}{10^{-3}} t)$$

$$x(n) = [2, 3, 4, 5, 4, 3, 2, 1, 0, 0, 0, 1, 2....]$$




Suppose we do not carry out any processing. Y[n] = x[n]. Are we able to regenerate the original signal?

## Digital to Analog Converter

$$x(n) = [2, 3, 4, 5, 4, 3, 2, 1, 0, 0, 0, 1, 2....]$$

| Voltage Range (Volts) | Number |
|---|---|
| 0-0.2 | 0 |
| 0.2-0.4 | 1 |
| 0.4-0.6 | 2 |
| 0.6-0.8 | 3 |
| 0.8-1.0 | 4 |
| 1.0-1.2 | 5 |
| 1.2-1.4 | 6 |
| 1.4-1.6 | 7 |
| 1.6-1.8 | 8 |
| 1.8-2.0 | 9 |

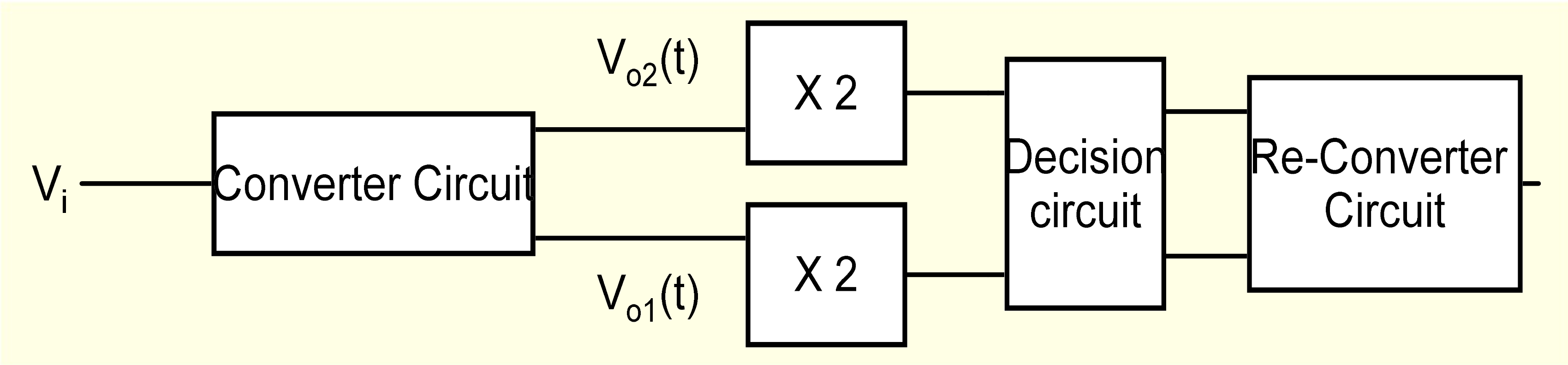
| Digital voltage or Number | New Analog Voltage |
|---|---|
| 2 | 2 x 0.2= 0.4 |
| 3 | 3 x 0.2=0.6 |
| 4 | 0.8 |
| 5 | 1.0 |
| 4 | 0.8 |
| 3 | 0.6 |
| 2 | 0.4 |
| 1 | 0.2 |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 1 | 0.2 |
| 2 | 0.4 |

## Digital to Analog Converter

| Voltage Range (Volts) | Number |
|---|---|
| 0-0.02 | 00 |
| 0.02-0.04 | 01 |
| 0.04-0.06 | 02 |
| - | - |
| - | - |
| - | - |
| 1.92-1.94 | 96 |
| 1.94-1.96 | 97 |
| 1.96-1.98 | 98 |
| 1.98-2.0 | 99 |

| Analog Voltage | Number | New Analog Voltage |
|---|---|---|
| 0.5 | 25 | 25 x 0.02 = 0.5 |
| 0.75 | 37 | 37 x 0.02 =0.74 |
| 0.93 | 46 | 0.92 |
| 1 | 50 | 1.0 |
| 0.93 | 46 | 0.92 |
| 0.75 | 37 | 0.74 |
| 0.5 | 25 | 0.5 |
| 0.25 | 12 | 0.24 |
| 0.067 | 03 | 0.06 |
| 0 | 00 | 0 |
| 0.067 | 03 | 0.06 |
| 0.25 | 12 | 0.24 |
| 0.5 | 25 | 0.5 |

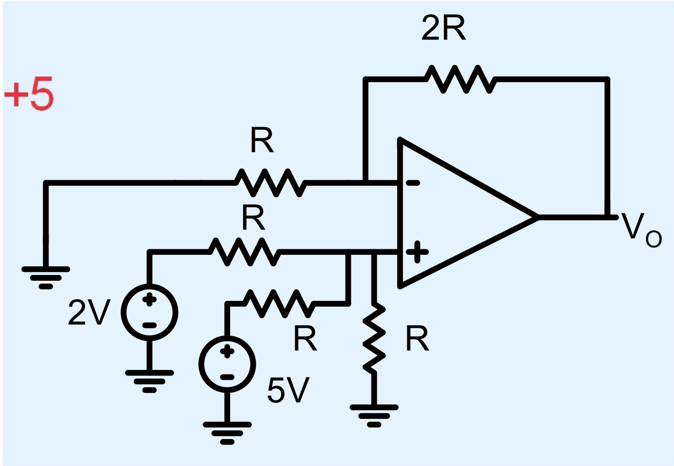
Processing of numbers in decimal system is cumbersome !




Circuits for processing numbers in binary system are much easier to implement

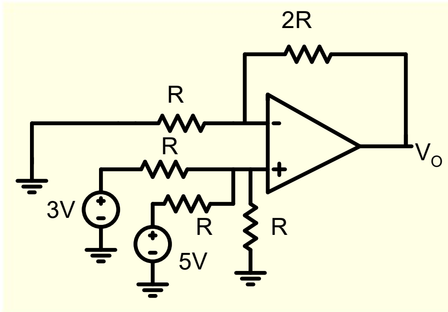
## **Processing of numbers in decimal system is cumbersome !**

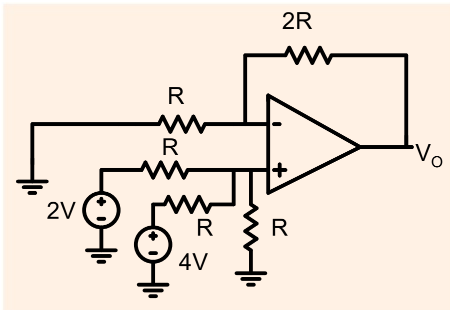
How do we add 2+5




How do we add 23+45?

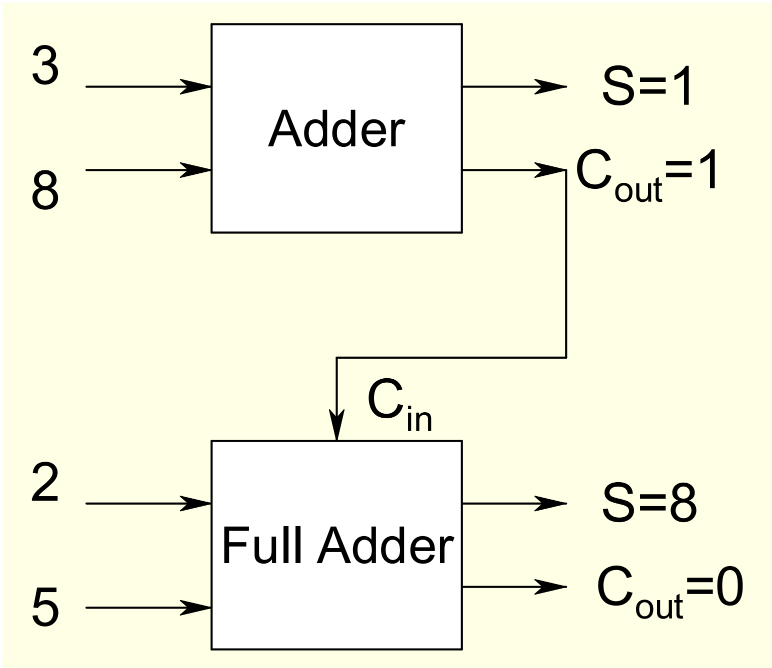
| | |
|---|---|
| 2 | 3 |
| 4 | 5 |
| 6 | 8 |



How do we add 23+58 ?

| | |
|---|---|
| 1 | |
| 2 | 3 |
| 5 | 8 |
| 8 | 1 |

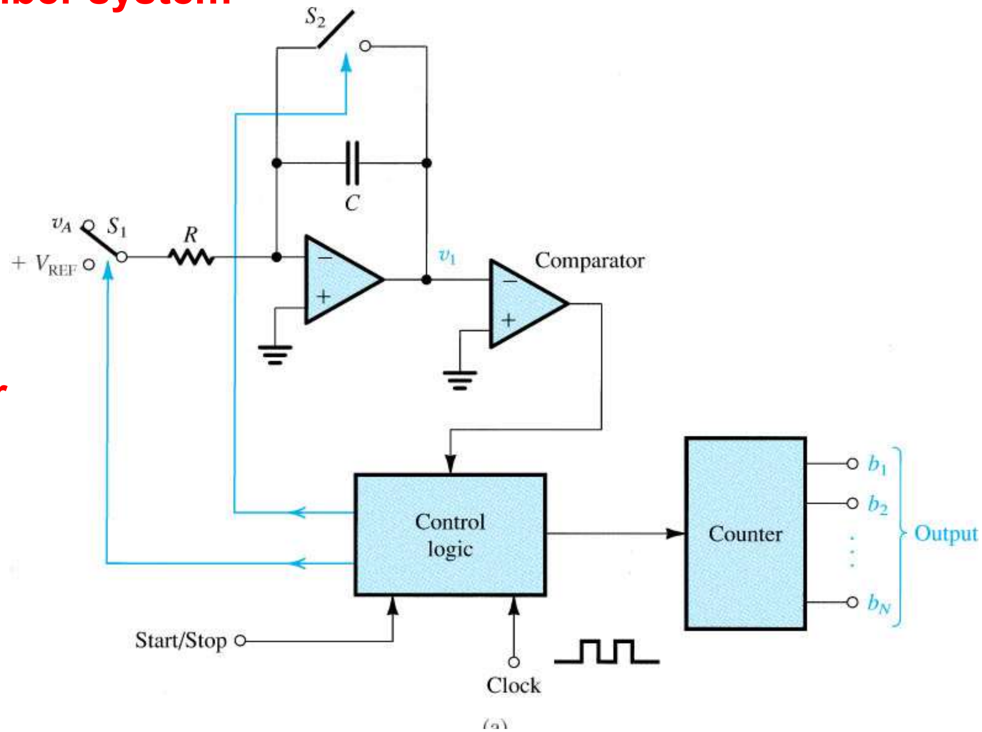



**It is not easy to design circuits to carry out this operations using decimal system**

**A Binary number system is more convenient !**

# Converter circuits for binary number system

## A/D converter




## D/A converter